APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: SWITCHING DATA PACKETS IN AN ETHERNET SWITCH

APPLICANT: RAMAKRISHNAN VENKATA SUBRAMANIAN

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 348189175 US

September 19, 2003

Date of Deposit

Switching data packets in an Ethernet Switch

Field of the invention

The present invention is related to data switches, particularly Ethernet switches.

Background of Invention

Conventionally Ethernet switches are provided in networks, such as LANs (Local Area Networks) including a plurality of computer units, each of which is associated with a respective MAC address. A data switch such as an Ethernet switch includes a plurality of ingress/egress ports and a switching fabric between them. Data packets arriving at one of the ports have a header containing the MAC address of the computer unit which transmitted the data packet and the MAC address of the destination of the data packet. The Ethernet switch gradually learns associations between incoming MAC addresses and ports, so that it can transmit the data packet to the port corresponding to the destination MAC address.

According to what data packets arrive at a switch, a data packet arriving at a given port may be queued until it can be transmitted. For this reason, the ports each include one or more queues for storing packets. The buffers for different ones of the ports are typically implemented in a single memory device, which may be a RAM memory.

This situation is illustrated in Fig. 1, which shows a shared memory 1 for storing data packets, in one or more queues for each of multiple ports. The packets stored may be identical to the data packets received at the switch, or they may be modified, e.g. with a different header. The memory 1 is structured to include sixteen packet buffers, lab lled PB1,...PB16. Typically

the number of packet buffers is much higher than this. Each of the packet buffers has the same size, referred to here as the PB length. Typically, this may be 256 bytes or 512 bytes.

Suppose that four packets of differing lengths are transmitted successively to the switch. These packets are illustrated schematically in Fig. 2 as packets 5,7,9,11, with each packet being shown with a different respective hashing scheme. The horizontal axis of Fig. 2 labels the lengths of the packets 5, 7,9, 11 in units of the PB length.

Conventionally, these packets will be stored in the memory 1 as illustrated in Fig. 3, where the hashing corresponds to Fig. 2 to indicate which of the packets are stored in which of the packet buffers. The first packet, packet 5, for example, which is slightly less than twice the PB length, is stored in PB1 and part of PB2. The second packet, packet 7, which is just more than one PB length, is stored in PB3 and part of PB 4. Note that, a given packet buffer does not store data from more than one packet. This means that the memory utilization is not efficient, and resources are wasted, and the performance of the switch can be degraded. For example, if a packet is only 4 bytes longer than twice the PB length, then three whole packet buffers will be required to store it. If the switch runs out of memory 1, the switch may have to refuse to accept new ones which reach it, and packets may be lost.

It is known to address this problem by attempting to optimise the sizes of the packet buffers, but these techniques are only successful to a limited degree, since certain of the packets may use these resources inefficiently.

Summary of the Invention

The present invention aims to provide a data switch which stores data packets more efficiently.

In general terms, the present invention proposes that, upon a data packet being received, the switch checks whether it can be stored efficiently in the packet buffers, and if the packet cannot be stored efficiently then a portion of the packet is stored in a separate memory and the remaining portion of the packet is stored in the packet buffers.

Specifically, one expression of the invention is a data switch having a plurality of ports, and a switching fabric for transferring data packets received at one of the ports to another of the ports specified by a header of the data packet, each of the ports being associated with one or more queues for data packets, the data switch further including:

a memory divided into packet buffers;

a plurality of registers;

a control unit for determining whether a data packet to be stored in one of the queues meets a criterion for efficient storage in the packet buffers, and otherwise dividing the data packet into a first portion which is stored in the packet buffers and a second portion which is stored in the register.

Preferably, the criterion for efficient storage is whether the length of the data packet is greater by more than a threshold than an integer multiple of the size of the packet buffers. Here the term "integer" is used to include zero as one of its possible realisations. In other words, the criterion is applied also to data packets which have a size less than that of a packet buffer: the criterion in that cas is simply whether the siz of the packet is, or is not, above the threshold.

The threshold value may be predetermined, or may be programmable. In the latter case, the data switch may include a memory storing the threshold value.

Brief Description of The Figures

Preferred features of the invention will now be described, for the sake of illustration only, with reference to the following figures in which:

Fig. 1 illustrates a packet storage mechanism used in a known data switch;

Fig. 2 illustrated four data packets input to this switch;

Fig. 3 illustrates how the packets of Fig. 2 are stored in the switch of Fig. 1;

Fig. 4 illustrated schematically a packet storage mechanism of a data switch according to the invention.

Detailed Description of the embodiments

The overall structure of packet storage mechanism in a data switch according to the invention is illustrated in Fig. 4. The mechanism includes a memory 1 having the configuration of Fig. 1, but in addition a control state machine 21 and a set of storage registers 23 and status registers 25. The memory 1 preferably functions as a shared memory for one or more queues of a plurality of ports. The control state machine 21 monitors data packets transmitted to the mechanism, and controls whether they are stored wholly in the memory 1 or partly in the registers 23.

In a first possible form of the switch, when a data packet arrives at the switch, it may initially be written the memory 1 in the manner described above in relation to Fig 2. That is, it is written to the lowest-numbered packet buffer(s) which are completely empty. In this case, it may fill on of more of the packets complet ly, and usually l aves a last packet buffer only partly full. The control

state machine 21 checks the last packet buffer, and determines the number of bytes contained there. If the number of bytes is low (e.g. low compared to a threshold) the control state machine 21 transfers the data from the last packet buffer to the set of registers 23, thus emptying that packet buffer. The threshold may be fixed (e.g. at 4 bytes, 8 bytes, 16 bytes or 32 bytes) or it may be programmable, i.e. controlled by a value stored in a further memory (not shown in Fig. 4).

In an alternative possible form of the switch, when a data packet arrives the control state machine 21 monitors it on-the-fly, and measures that the data packet is larger than an integer multiple of the packet buffer size by an overflow amount less than the threshold. If so, it partitions the data packet into a first portion which is stored in the packet buffers (and which preferably has a size which is an integer multiple of the packet buffer size) and a second portion which is transmitted directly to the registers 23 for storage without being written into the memory 1. This means that the control state machine 21 does not have to copy data from the packet buffer memory 1 to the registers 23.

Both forms of the invention achieve the same result. For example, the result of transmitting the data packets 5, 7, 9, 11 in turn to the packet storage mechanism of the invention is illustrated in Fig. 4. The data packet 5 is less than twice the size of the packet buffers, and thus is efficiently stored in the packet buffers PB1 and PB2, as in the prior art data switch of Fig. 3. However, the control state machine 21 determines that the packet 7 is more than one packet buffer length by an amount which is less than the threshold, and therefore divides the packet 7 into a first portion which is one packet buffer length and a second portion. The first portion is stored in the packet buffer PB3, and the second portion 27 is stored in th registers 23. Similarly, the third packet 9 is divided into a first portion which is stored in the packet buffer

PB4 and a second portion 29 which is stored in the registers 23. The control state mechanism 21 determines that the final packet 11 has a length slightly less than three times the packet buffer length, and accordingly stores it in the three packet buffers PB5, PB6, PB7. Thus, comparing Figs. 3 and 5, the embodiment is able to store the packets using seven packet buffers rather than nine.

The status registers are used to record which of the storage registers 23 have been used, and which packets they relate to. This information is used by the control state machine 21 when deciding which of the storage registers 23 should be used to store the second portions of new data packets. Specifically, when a new packet is to be stored, and when it is determined that a second portion of the packet is to be stored in the storage registers 23, the control state machine 21 uses the states stored in the status registers 25 to decide which of the storage registers 23 should be used to store that second portion. The status registers 25 are then updated.

The control state machine 21 is also operative to extract information from the packet buffers and the memory, in response to a read instruction. When a read instruction is received the control state machine 21 examines the status registers 25 to determine whether any of the storage registers 23 stores a part of the packet, and if so extracts the second portions of the packets from the storage registers 23. The control state machine 21 then transmits the first and second portions of the packets out of the packet storage mechanism in the correct order, and updates the status registers to indicate that the packet is no longer stored in the packet storage mechanism.

Note that the memory 1 is typically implemented as a RAM memory, which is a term which, as used in this field, does not include registers, such as the storage registers 23 and status registers 25.

A skilled reader will appreciate that other portions of the data switch than the packet queues may be implemented by any known system, such as according to the Ethernet standard.

Although only a single embodiment of the invention has been illustrated, the invention is not limited in this respect, and many variations are possible within the scope of the invention as will be clear to a skilled reader.